Serious Game and Students' Learning Motivation: Effect of Context Using Prog&Play

Mathieu Muratet, Elisabeth Delozanne, Patrice Torguet, Fabienne Viallet

Serious game and students' learning motivation: effect of context using Prog&Play

Mathieu Muratet[1], Elisabeth Delozanne[1],
Patrice Torguet[2,4], and Fabienne Viallet[3,4]

[1] LIP6, Université Pierre et Marie Curie, 4 place Jussieu,
75005 Paris, France, {mathieu.muratet,elisabeth.delozanne}@lip6.fr
[2] IRIT, patrice.torguet@irit.fr
[3] UMR EFTS, fabienne.viallet@univ-tlse3.fr
[4] Université Paul Sabatier, 118 route de Narbone, 31400 Toulouse, France

Abstract. This paper deals with an analysis of a large-scale use of Prog&Play[5], a game-based learning environment specially designed to teach the basics of programming to first year university students. The study relies mainly on a motivation survey completed by 182 students among 258 who used the serious game for 4 to 20 hours in seven different university settings. Our findings show that the students' interest for Prog&Play is not only related to the intrinsic game quality, it is also related to the teaching context and mainly to the course schedule and the way teachers organize sessions to benefit from the technology.

Key words: Serious games, programming, algorithms, motivation

1 Introduction

Many studies report the growing disinterest of students in developed countries for science in general and for computer science in particular [1, 12]. To face an urgent need to improve the level of understanding of computer science as an academic and professional field, many countries are implementing curricula to teach computational thinking[6] [2, 14]. At the same time, an important effort is underway to define pedagogical approaches that will make thinking in terms of computer science more accessible and attractive to all students. These approaches include international competitions between schools[7] or between countries[8]. Other studies show that video games are a successful way to increase student motivation by making learning fun. For example, they support problem-based learning and experiential learning, and they provide immediate feedback, enabling students to self-assess their actions or strategies [11]. The work presented here is a contribution to that field of research. Our basic assumptions are (i) that video games

[5] Prog&Play is an open source serious game freely downloadable at http://www.irit.fr/ProgAndPlay/index_en.php
[6] Programming Skills Development, http://pskills.ced.tuc.gr/
[7] Bebras, http://www.bebras.org/en/welcome
[8] International Olympiad in Informatics, http://www.ioinformatics.org/

are exciting for students, and (ii) that they can also provide a good context to embed the teaching of computer programming.

Our project, called Prog&Play, aims at increasing students' motivation for learning the basics of programming by writing programs to manipulate the units of a real-time strategy game (RTS). If students implement efficient strategies, they will improve their chance to defeat their enemies and to win missions. In a previous paper, we detailed the design, implementation and evaluation of Prog&Play [10]. In this paper, we investigate how students' motivation is related to the teaching context. First, we discuss background and related work. Then, we present the different experiments we conducted to test Prog&Play with 258 undergraduate students and 20 teachers in different university settings. Finally, we analyse the results to outline guidelines for a successful use of Prog&Play and suggest further avenues of research.

2 Background and related work

A popular use of a game-based learning approach to teach programming is asking students to implement their own video game. Chen and Cheng [3] use C++ to enable students to build a small-to-medium scale interactive computer game in one semester. Tools like Scratch [9] or Alice2 [7] are used to make first programming experiences more engaging.

Another approach consists in using programming games where the player has to write computer programs or scripts in order to control the actions of game units. In Colobot[9], users colonise planets using robots that they program in a specific object-oriented language similar to C++. Other projects do not use a storytelling approach but rely on competition to increase motivation. Robocode [6] is a Java programming game, where the goal is to program a robot tank to fight against other tanks programmed by other players. Other such games are Gun-Tactyx[10] using the SMALL language or Robot Battle[11] using a specific script language.

In the Prog&Play project, to ensure contextual learning, we use a storytelling approach where students have to carry out missions as in Colobots, but it is also possible to organize competition between students' programs. Moreover, to adapt to different teaching contexts, Prog&Play provides a large choice of programming languages to command game units: Ada, C/C++, Compalgo, Java, OCaml and Scratch. Prog&Play relies on three basic principles: (i) learners program the game units with simple programs involving functions from a teacher customizable library; (ii) learners see the results of their programs in the game context where they influence the game results; and (iii) learners' engagement is based on storytelling or competition. Our storytelling approach embeds the pedagogical objectives in different missions to be carried out. While our competitive approach motivates students to improve their programs in order to beat other players.

[9] Colobot, `http://www.ceebot.com/colobot/index-e.php`
[10] Gun-Tactyx, `http://apocalyx.sourceforge.net/guntactyx/`
[11] Robot Battle, `http://www.robotbattle.com/`

3 Evaluation

Our goals in designing and implementing Prog&Play were to produce benefits in terms of students' motivation and curricular-specific learning outcomes. As Prog&Play was not used as a standalone learning environment, but was used in different actual university settings, it was difficult to detect the learning outcomes due to Prog&Play or to the teachers' specific pedagogical strategy. To evaluate Prog&Play, (i) we used an iterative and collaborative design and evaluation method involving teachers in order to understand how they implement Prog&Play in the different introductory programming courses they were responsible for, and (ii) we delivered a post questionnaire to students. Our research question was: Is there a relationship between students motivation and the teaching context in which Prog&Play was used and which context is more beneficial?

3.1 Usage settings and participants

We studied usage of Prog&Play in seven different settings (noted S1 to S7) involving 258 students and 20 teachers. Teachers organized the pace, schedule and evaluation of students work with respect to their institutional constraints. No member of the Prog&Play design team was involved as a teacher in S4, S6 and S7. In S4 and S5, Prog&Play practice sessions were mandatory and integrated within the regular course, while in the other settings, it was used in addition to the regular course. In S6 and S7, both teachers especially designed courses called "Learning with Information Technology" and "Learning differently" to investigate new pedagogical approaches with Prog&Play in two different universities.

In every setting, Prog&Play was already installed on computers and a teacher was in the room presenting the teaching concepts, the environment, the library and providing help when asked by students. Only in the 6th setting, after 5 sessions with a teacher, students had to complete the game at home with the teacher's or peers' e-mail support to install the game or to debug their programs.

3.2 Materials

To collect information on students' motivation, we designed a questionnaire using the hierarchy of players' needs proposed by Siang and Rao [15] and Greitzer *et al.* [5]. These authors adapted Maslow's original hierarchy of needs to define seven criteria to be fulfilled to motivate players in a game: *rules need* (need 1); *safety need* (need 2); *belongingness need* (need 3); *esteem need* (need 4); *need to know and understand* (need 5); *aesthetic need* (need 6); and *self actualization need* (need 7). Following these authors, our assumption was that the degree of satisfaction within this hierarchy of needs was a significant indicator of motivation.

3.3 Results and analysis

We considered only questionnaires that were fully completed by students (S1: 13/15; S2: 23/35; S3: 16/16; S4: 29/60; S5: 91/99; S6: 10/18; S7: 0/15; Total:

Table 1. Usage of Prog&Play in seven different settings and global satisfaction.

	N	Language, Teaching context and Time spend on game	SR*
S1	15	Compalgo, Workshop apart from regular teaching, 5 * 1h30	4.6/10
S2	35	C, Practice for failing students in addition to regular teaching, 3 * 1h30	3.8/10
S3	16	Java, Workshop apart from regular teaching, 3 * 1h30	4.1/10
S4	60	C, Compulsory practice sessions for every student, 5 * 1h30	2.7/10
S5	99	OCaml, Compulsory practice sessions for every student, 2 * 2h	2.6/10
S6	18	C, Workshop part of a regular IT course, 6 * 2h + homework	6.3/10
S7	15	C, Workshop, regular teaching designed for failing students, 5 * 2h	unreported

* Satisfaction Rate

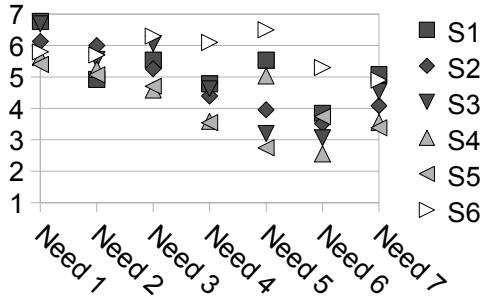

Fig. 1. Mean satisfaction for each player's need in each setting

182/258). We compared (Table 1) students' satisfaction rates in each setting by means of Likert items on the seven need levels. Only a quarter of the students were satisfied in S4 and S5, where Prog&Play practice sessions were mandatory in the regular course schedule. In S1, S2 and S3 where Prog&Play was used in addition to the regular course schedule (as a workshop or practical exercices for students with low grades), the rate of satisfied students was 4 students out of 10. And in S6 where Prog&Play is used as a project assignment, the rate rose to 6 out of 10.

These results suggest that Prog&Play is better implemented within projects, workshops or supplementary practical sessions. We conjecture that Prog&Play is not a game that *teaches* computer programming basics, but it provides a micro-world [13] where students can explore the effects of their different programming constructs and learn from the feedback given by the micro-world. Students use taught programming concepts in an appealing context (RTS) whereas, in regular teaching, they are required to use them in a mathematical context (and they are evaluated using them in such an abstract context).

In addition, we hypothesize that the schedule is an important motivation factor. In regular teaching (S2, S3, S5), teachers split the game scenario into different sessions to fit the pace of programming concepts being introduced, whereas the gameplay would require a more continuous gameflow [4] built on the progression of the missions. Moreover, teachers urged students to finish on time by giving them a solution, while in a normal game session, players often enjoy finding solutions on their own. The course agenda is easier to adapt when the game is used as an add-on to the existing teaching materials (S6, S7).

Figure 1 studied the mean satisfaction for each need level of Siang and Rao's hierarchy in each setting. It shows a greater dispersion of answers on need 5 (Need to know and understand). Need 5 is defined as the necessity for the player to discover new game elements in order to reuse them in future parts of the game. In Prog&Play, this need is satisfied through discovering new units, with their own features, as well as new ways to command units (through programming constructs). This reinforces our hypothesis that the discovery part is important for motivation and learning and requires a suffisiant exploration time for players. Satisfaction for need 5 seems therefore highly tied to the time allocated to the game in the teaching agenda.

4 Conclusion

In this paper, we have briefly described Prog&Play, a game-based learning environment, and presented data collected when it was introduced in different university settings. The questionnaires collected from students suggest a clear influence of the teaching setting on students' motivation: a workshop or a project based course in addition to a traditional introductory course, is clearly more beneficial than just plugging Prog&Play sessions within a traditional course. Furthermore, we identified that giving enough time to students to discover the game world and rules is a key feature to improve game understanding and therefore to increase their motivation.

Data collected suggest that, using a serious game only as an illustration tool inside regular teaching doesn't seem to be very beneficial to motivation. In S6 and S7 where there were less time constraints and where the game flow was continuous, students enjoyed advantages inherited from video games: they carried out actions within the game and observed their effects on the game to improve their knowledge of programming constructs. The opportunity for students to carry out useless, redundant or incorrect actions within a serious game providing feedback [16] is fundamental to catch players attention and to allows them to understand programming concepts deeply. A student in setting 6 described very well the motivation induced by exploring the game: "*The solution of the seventh mission took a long time to be achieved. Lots of ideas were considered and left unused. In the end, hundreds of code lines were written. I saw my army destroyed many many times. But, each attempt brought me closer to victory and kept me in suspense. Due to this suspense I completed this mission*".

Acknowledgments

We thank Rebecca Freund, Thomas Joufflineau and John Wisdom for helping with English, and teachers from Universities and IUT of Toulouse and Paris who used Prog&Play in their course.
The original publication is available at www.springerlink.com

References

1. ACM, IEEE-CS: Computer Science Curriculum 2008: An Interim Revision of CS2001. ACM Press. and IEEE Computer Society Press., New York, (2008)
2. Archambault, P.: Un enseignement de la discipline informatique en Terminale scientifique. In: DIDAPRO 4 - Dida&STIC, pp. 205–212, (2011)
3. Chen, W.-K., Cheng, Y. C.: Teaching Object-Oriented Programming Laboratory With Computer Game Programming. In: Education, IEEE Transactions on, 50(3), pp. 197–203, (2007)
4. Csikszentmihalyi, M.: Flow - The Psychology of optimal Experience. Harper Perennial, (1991)
5. Greitzer, F. L., Kuchar, O. A., Huston, K.: Cognitive science implications for enhancing training effectiveness in a serious gaming context. J. Educ. Resour. Comput., 7(3), art. 2, (2007)
6. Hartness, K.: Robocode: using games to teach artificial intelligence. J. of Comput. Sciences in Colleges, 19(4), pp. 287–291, (2004)
7. Kelleher, C., Cosgrove, D., Culyba, D., Forlines, C., Pratt, J., Pausch, R.: Alice2: Programming without Syntax Errors. In: 15th annual symposium on the User Interface Software and Technology, (2002)
8. Leutenegger, S., Edgington, J.: A games First Approach to Teaching Introductory Programming. In: 38th SIGCSE technical symposium on Computer science education, 39(1), pp. 115–118, (2007)
9. Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., Resnick, M.: Scratch: A Sneak Preview. In: 2nd International Conference on Creating Connecting, and Collaborating through Computing, pp. 104–109, (2004)
10. Muratet, M., Torguet, P., Viallet, F., Jessel, J.-P.: Experimental feedback on Prog&Play: a serious game for programming practice. In: Computer Graphics Forum, Eurographics, 30(1), pp. 61–73, (2011)
11. Oblinger, D.: The Next Generation of Educational Engagement. In: J. of Interactive Media in Education, (2004)
12. Papastergiou, M.: Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. In: J. Comput. Educ., 52(1), pp. 1–12, (2009)
13. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books: New York, (1980)
14. Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cuniff, D., Boucher Owens, B., Stephenson, C., Verno, A.: CSTA K-12 Computer Science Standards. Revised 2011, CSTA Standards Task Force, (2011)
15. Siang, A. C., Rao, R. K.:Theories of learning: a computer game perspective. Multimedia Software Engineering, pp. 239–245, (2003)
16. Thomas, P., Yessad, A., Labat, J.-M.: Petri nets and ontologies: tools for the "learning player" assessment in serious games. In: Advanced Learning Technologies, pp. 415–419, (2011)